Online Sales Prediction

Nick Pistiolis

June 10, 2015

Introduction

The prediction of product sales, based on known features for example campaign costs, is the ultimate goal for every business. Many firms collect data that can be used for statistical analysis to achieve such predictions. In this report we are going to predict next year online sales based on the previous year online sales of many products.The regression model we are going to use for this purpose is randomForest.**The code is in R**

Data

The data contains twelve months sales (twelve first columns of training data). Each row is a different product. All the other columns are quantitative variables and categorical variables. In this report we are going to treat all variables as quantitative variables because many categorical variables have many levels. The data are training and test. Test data do not have the fisrst twelve columns, these are the columns we are going to predict. Finally there are two columns that represent the date when the product was announced and when it released.

Preprocessing the data

Load the data.

```
training = read.csv("TrainingDataset1.csv", na.strings="NaN")
test = read.csv("TestDataset.csv", na.strings="NaN")
```

Fill NA's with the mean of each column.

```
for (i in 1:ncol(training)) {
  for(y in 1:nrow(training)){
    if (is.na(training[y,i]))
    {training[y,i]<-as.integer(mean(training[,i],na.rm=TRUE))}
  }
}
```

The dates are not in a correct format. They are just the number of days with the start date unknown. So we subtract one from another, to find the period from which the product is announced and released.

```
date<-subset(training, select=c(Date_1,Date_2))
date<-date[,1]-date[,2]
```

```
training<-cbind(training,date)
training<-subset(training, select=-c(Date_1,Date_2))
```

Because test product sales do not have the actual sales to compare them with the predicted sales,we seperate randomly training data into "training1" and "test1" data to see the accuracy of our regression model.

```
smp_size <- floor(0.80 * nrow(training))
set.seed(13)
train_ind <- sample(seq_len(nrow(training)), size = smp_size)
training1 <-training[train_ind, ]
test1 <-training[-train_ind, ]
```

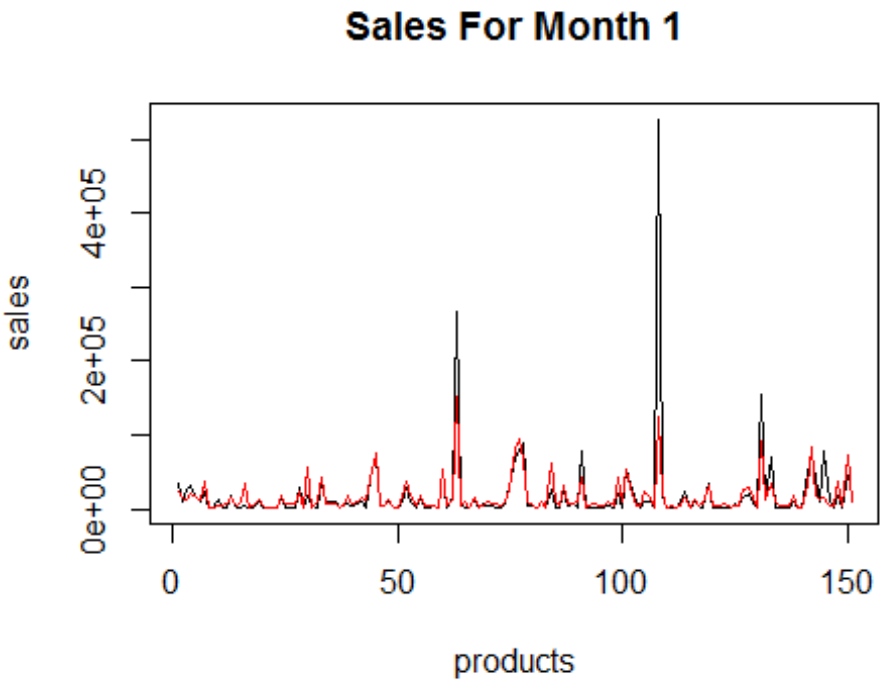
Main anlysis

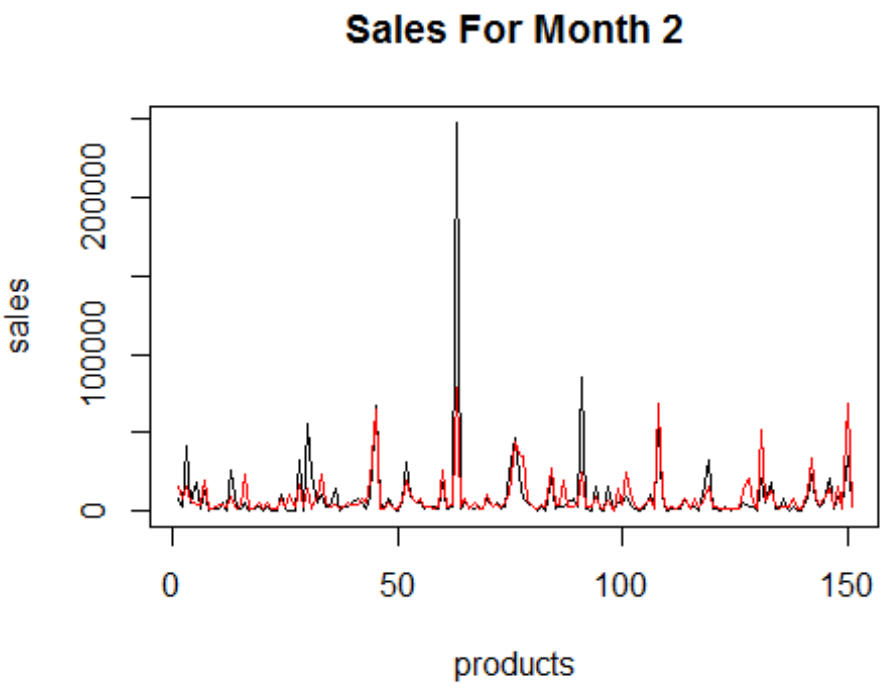
We are using randomForest as a regression model.

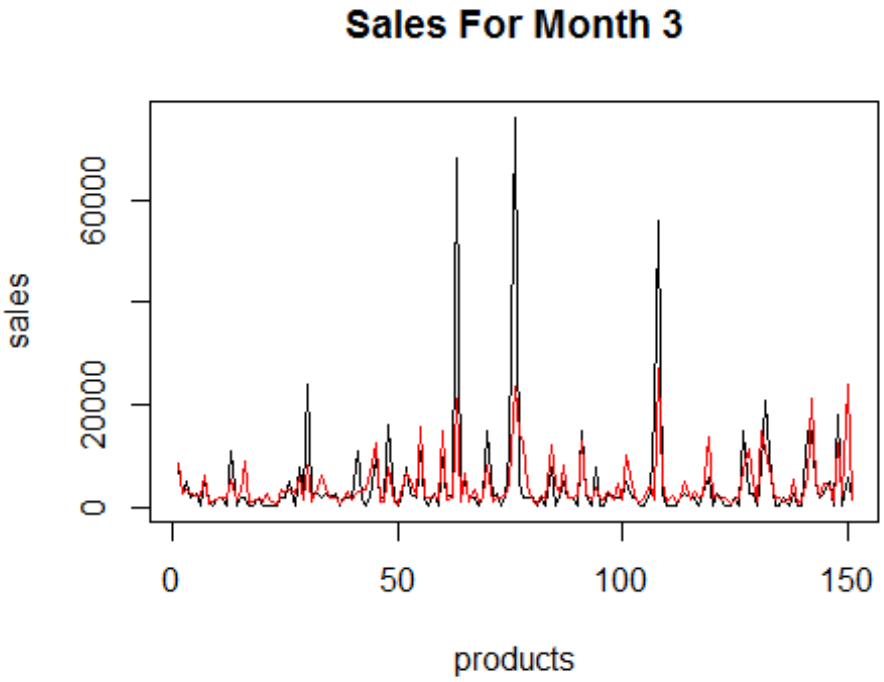
```
library(randomForest)

## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.

predict_rf<-data.frame(matrix(NA, nrow = nrow(test1), ncol = 12))
newdata_rf<-data.frame(test1[,13:ncol(test1)])

for(var in 1:12){
  rf <-randomForest(training1[,13:ncol(training1)],training1[,var])
  predict_rf[,var]<-predict(rf,newdata=newdata_rf)
}
```

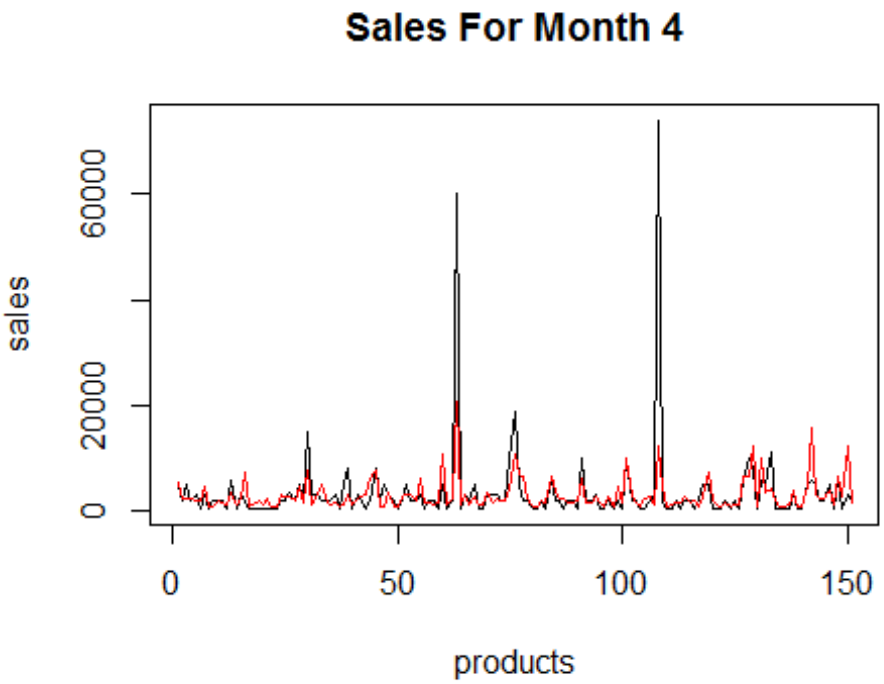
We plot the predictions(red line) and compare them with the actual sales (black line)

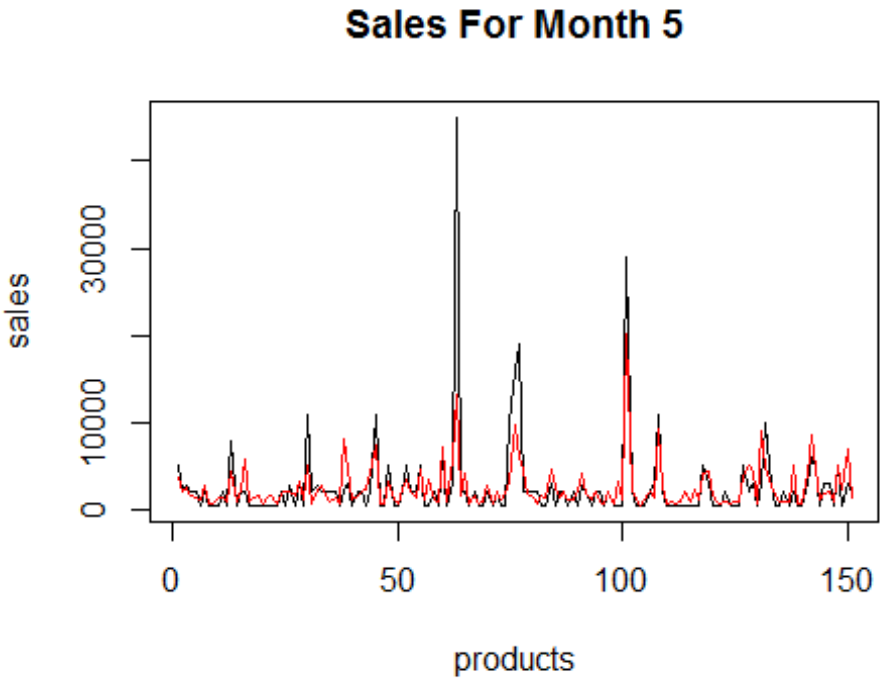
```
for (i in 1:12){

plot(1:nrow(test1),test1[,i],"l",xlab="products",ylab="sales",main=paste("Sal
es For Month",i))
  lines(1:nrow(test1),predict_rf[,i],col="red")
}
```

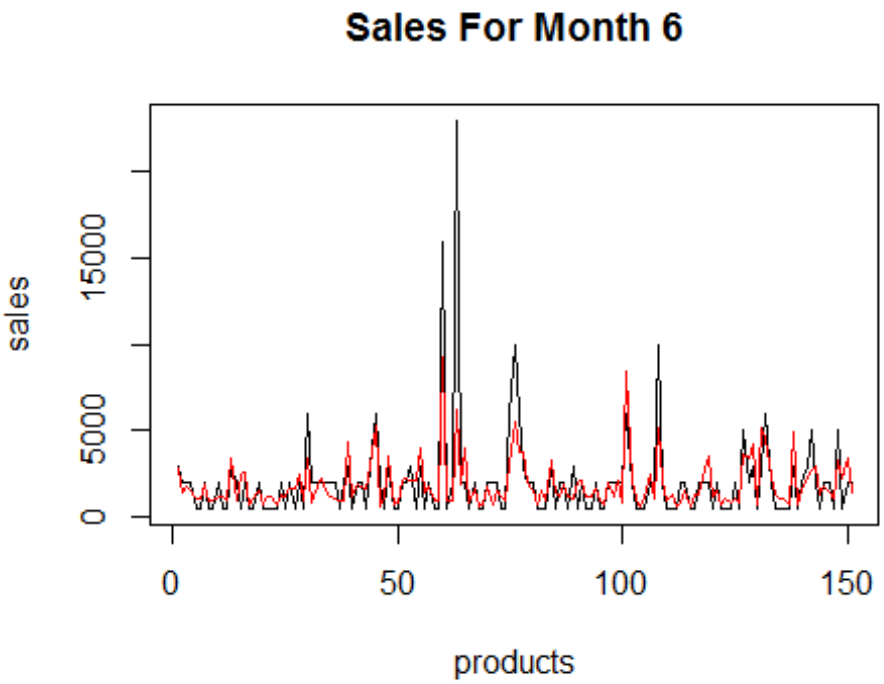

Sales For Month 1
sales
0e+00
2e+05
4e+05
0
50
100
150
products

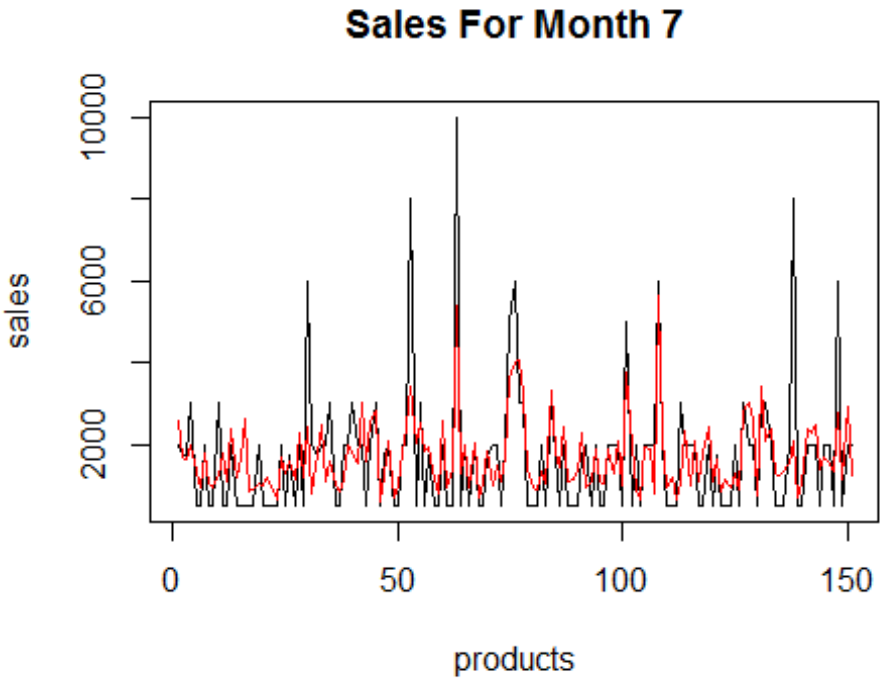

Sales For Month 2
sales
0
100000
200000
0
50
100
150
products

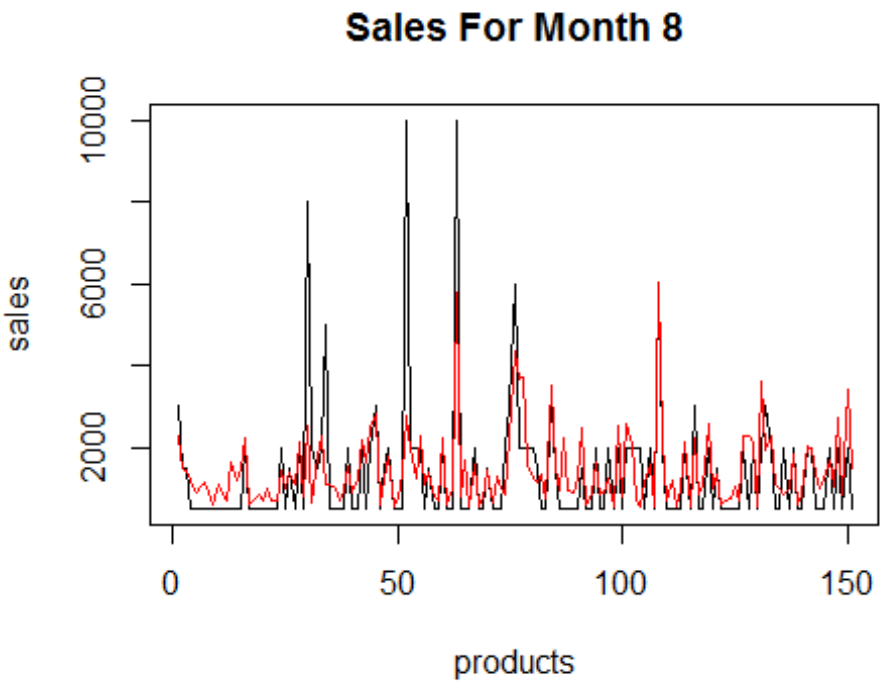

Sales For Month 3
sales
0
20000
60000
0
50
100
150
products

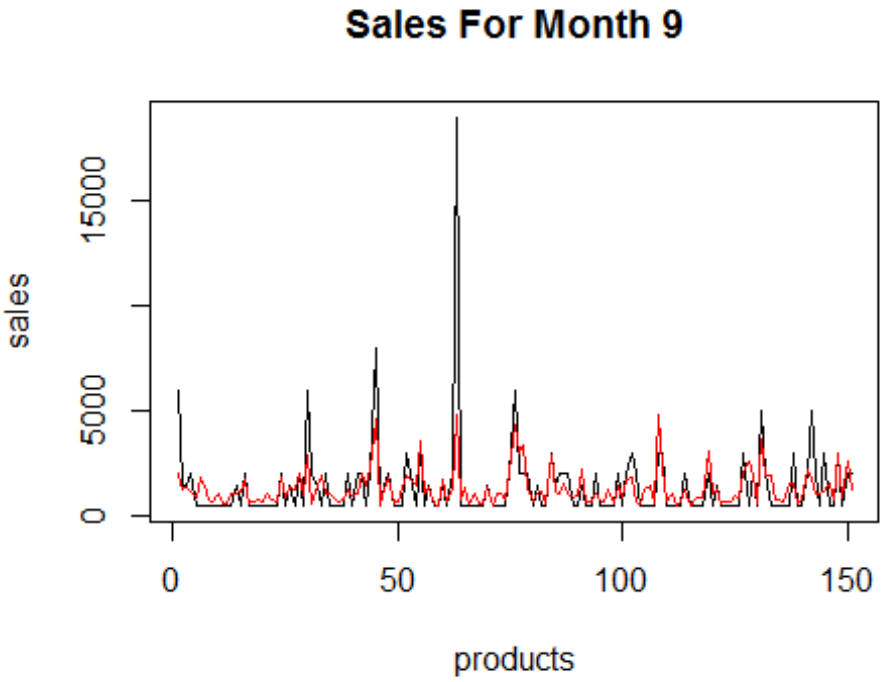

Sales For Month 4
sales
0
20000
60000
0
50
100
150
products

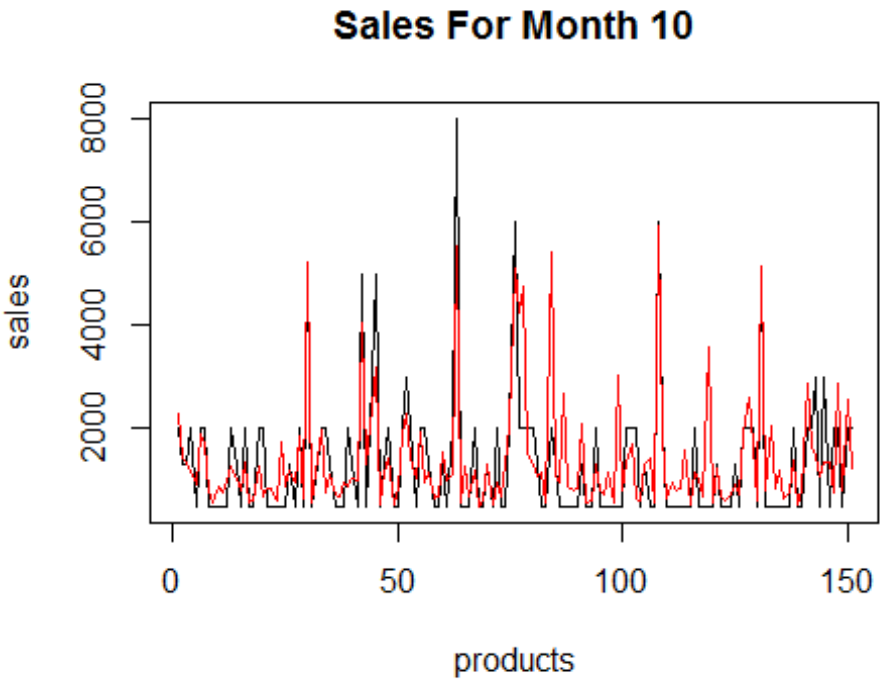

Sales For Month 5
sales
0
10000
30000
0
50
100
150
products

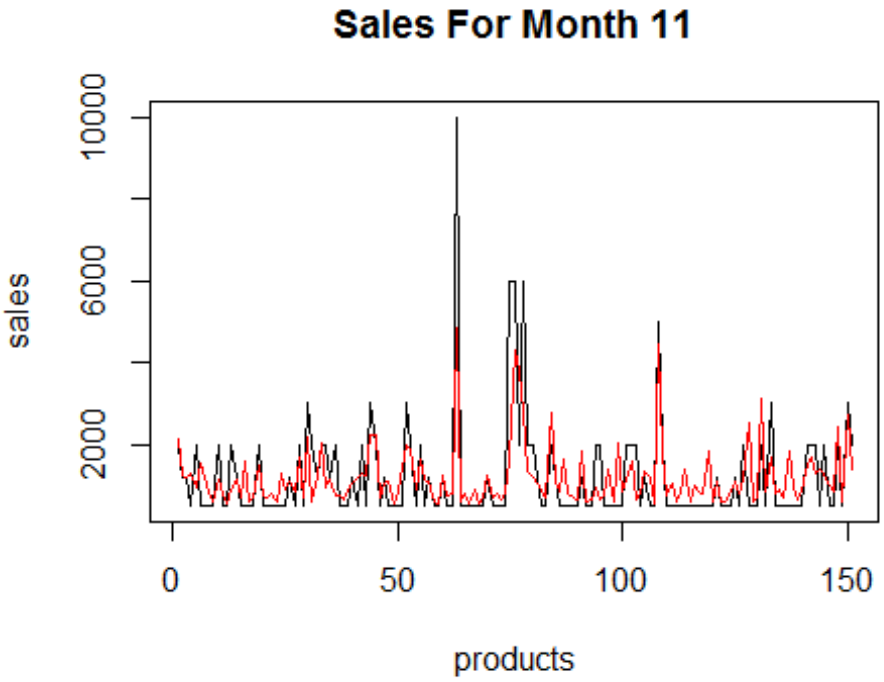

Sales For Month 6
sales
0
5000
15000
0
50
100
150
products

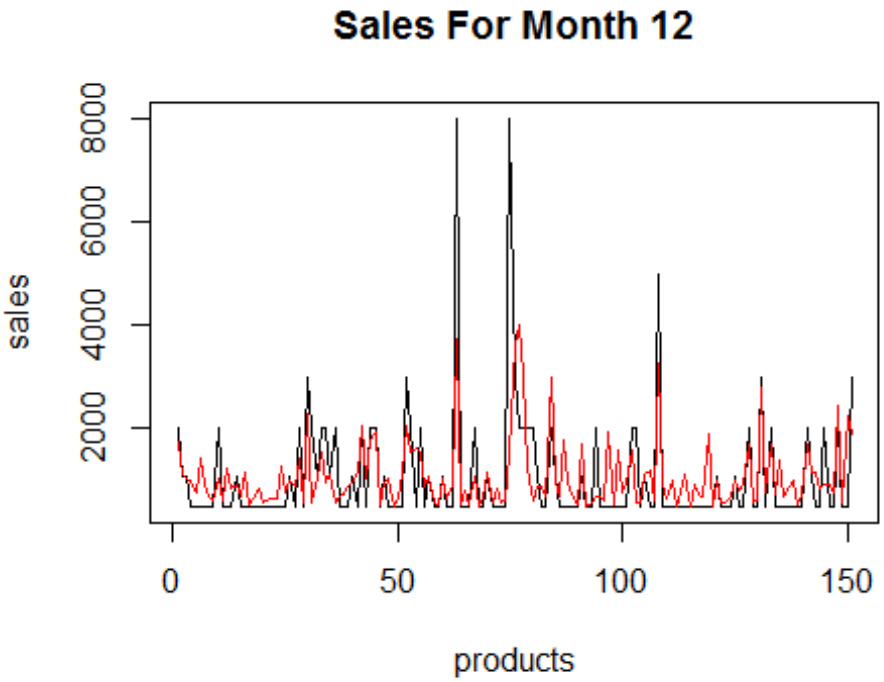

Sales For Month 7
sales
2000
6000
10000
0
50
100
150
products


Sales For Month 8
sales
2000
6000
10000
0
50
100
150
products


Sales For Month 9
sales
0
5000
15000
0
50
100
150
products


Sales For Month 10
sales
2000
4000
6000
8000
0
50
100
150
products





As an error metric, we use Root Mean Square Logarithmic Error (RMSLE) which was used in the Kaggle online sales competition. **The first place had error 0.56** . We use that metric to see if we are close to that number. The smaller RMSLE means better prediction. We calculate RMSLE for each month seperately and then we take the mean.

```
library(Metrics)

rmsle<-vector("numeric",12)

for (i in 1:12){
  rmsle[i]<-rmsle(test1[,i],predict_rf[,i])
}
#RMSLE
mean(rmsle)
```

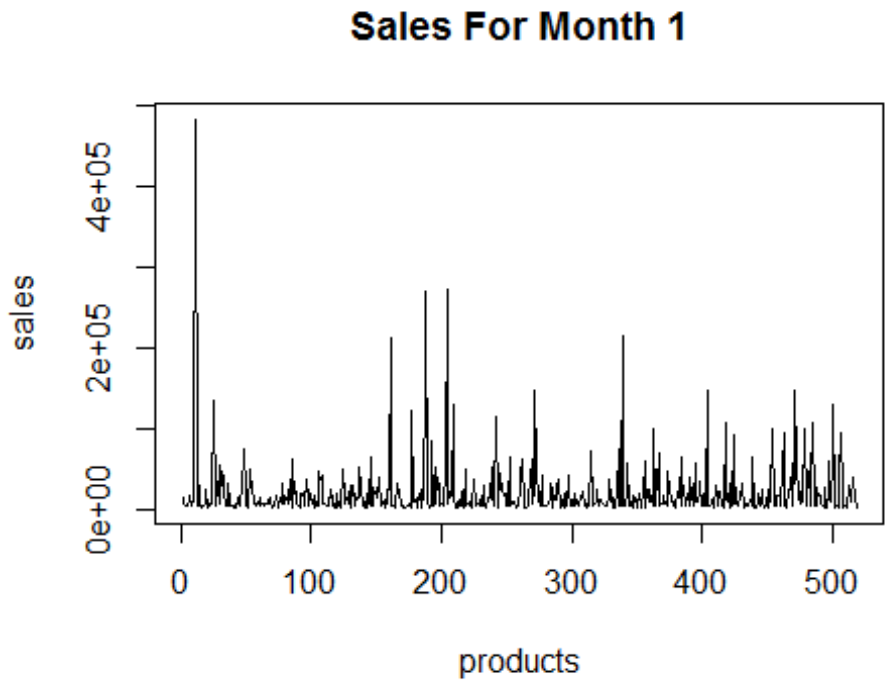
```
## [1] 0.6638555
```

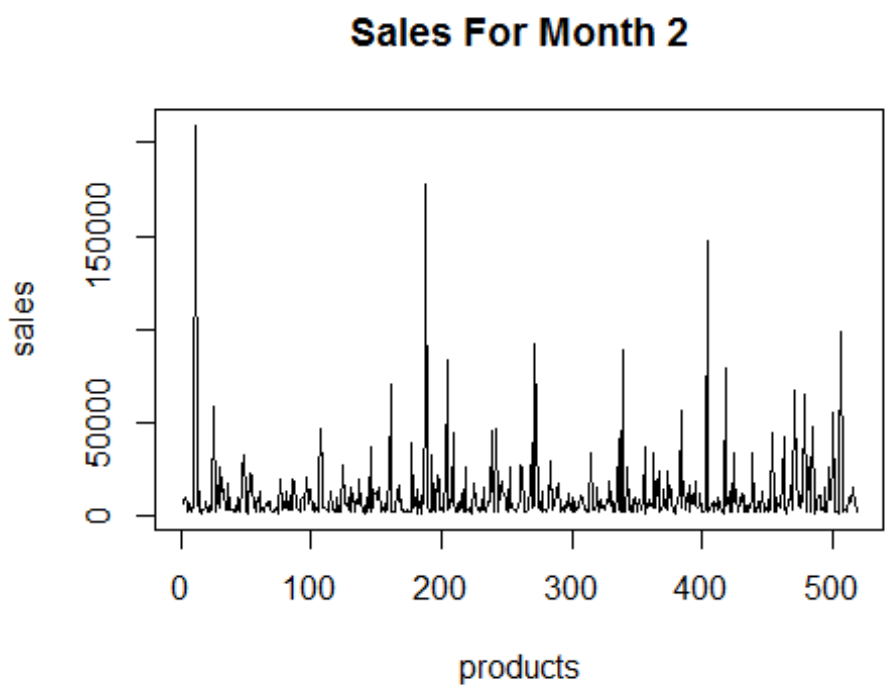
As we can see **RMSLE=0.66**. This value changes if we select another "training1" and "test1" from training data. We must also specify that we had to predict sales for different products(test1) from those we used for training data (training1).**The real test is to predict products sales (test) by having as training data all product sales, which we do next. The error of this prediction is expected to be lower because training data are complete. But we don't have the actual sales to compare the results.**

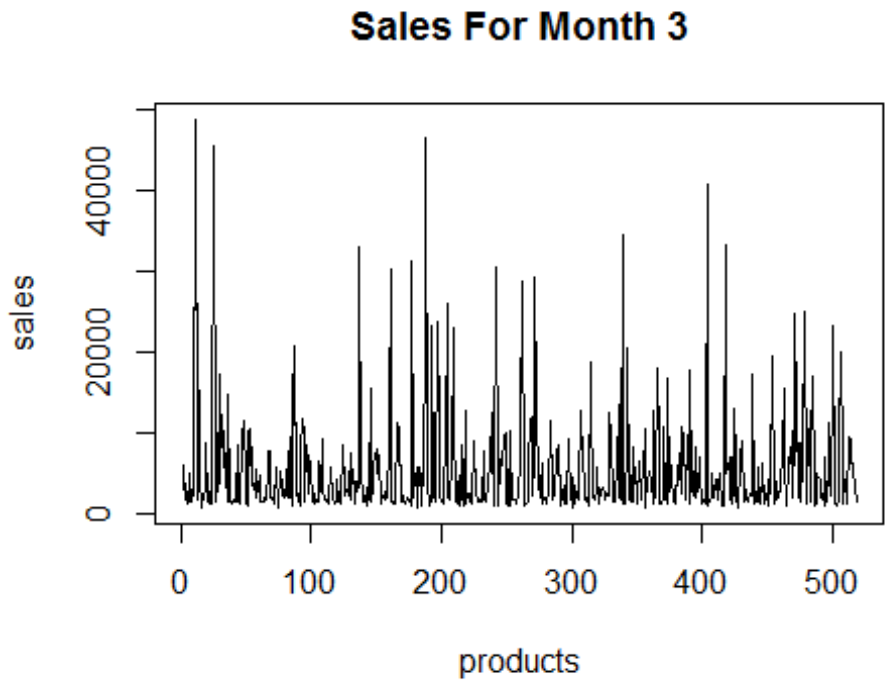
Main prediction

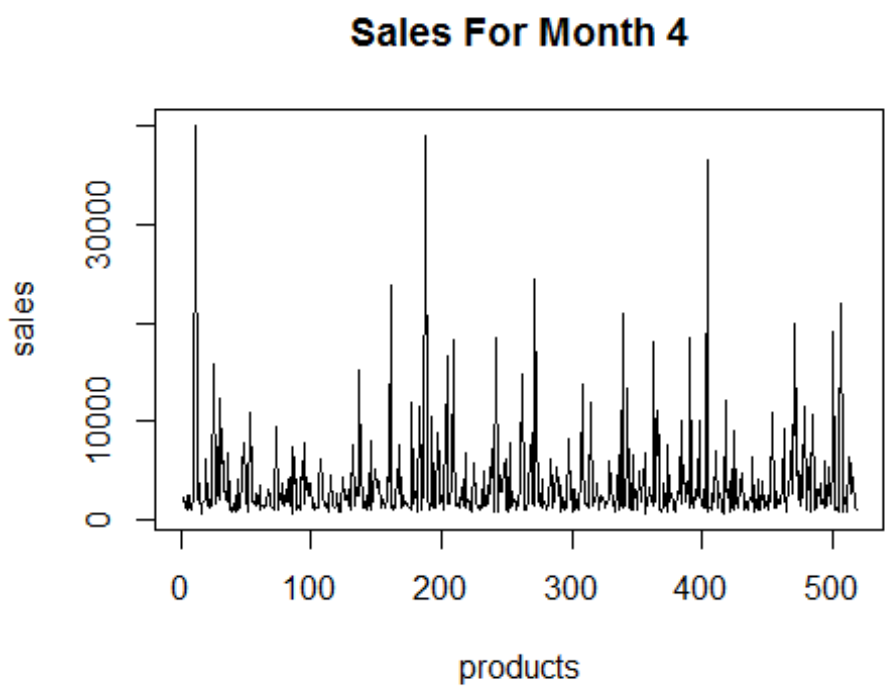
Bellow we do the same preprocessing and main analysis for test data.

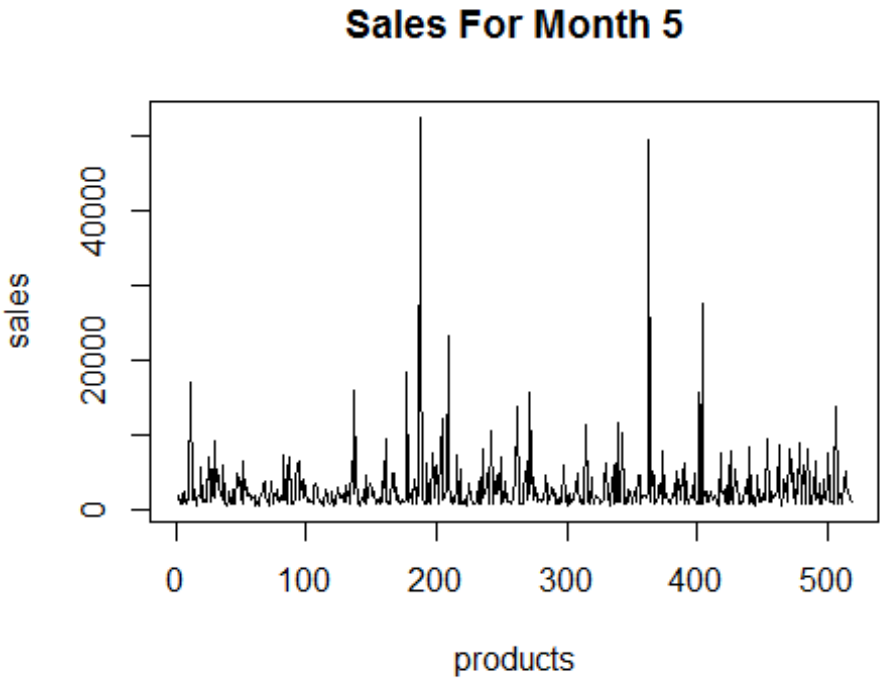
```
#Preprcessing test data
for (y in 1:(ncol(test))) {
  for(x in 1:nrow(test)){
    if (is.na(test[x,y]))
    {test[x,y]<-as.integer(mean(test[,y],na.rm=TRUE))}
  }
}

test<-test[,2:ncol(test)]
date<-subset(test, select=c(Date_1,Date_2))
date<-date[,1]-date[,2]
test<-cbind(test,date)
test<-subset(test, select=-c(Date_1,Date_2))

#Main prediction
predict_rf_f<-data.frame(matrix(NA, nrow = nrow(test), ncol = 12))

for(var in 1:12){
  rf<-randomForest(training[,13:ncol(training)],training[,var])
  predict_rf_f[,var]<-predict(rf,newdata=test)
}

#Plots
for (i in 1:12){
plot(1:nrow(test),predict_rf_f[,i],"l",xlab="products",ylab="sales",main=past
e("Sales For Month",i))
}
```

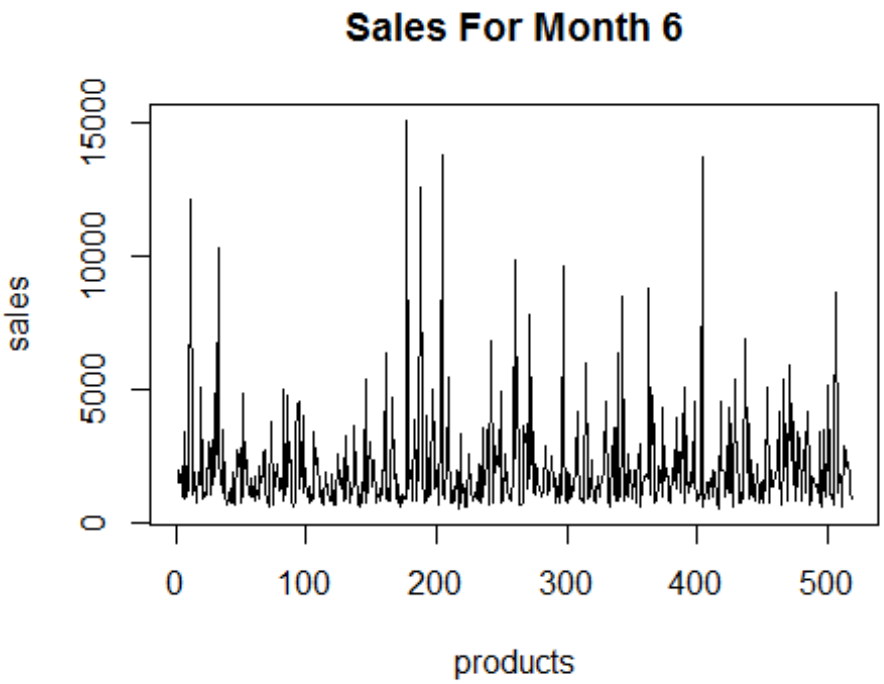

Sales For Month 1
sales
0e+00
2e+05
4e+05
0
100
200
300
400
500
products

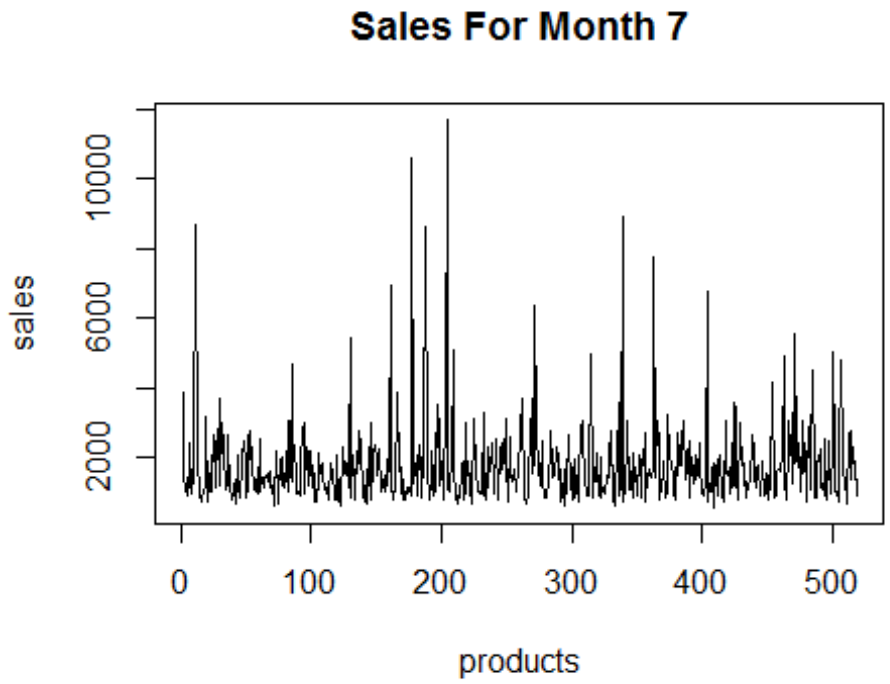

Sales For Month 2
sales
0
50000
150000
0
100
200
300
400
500
products

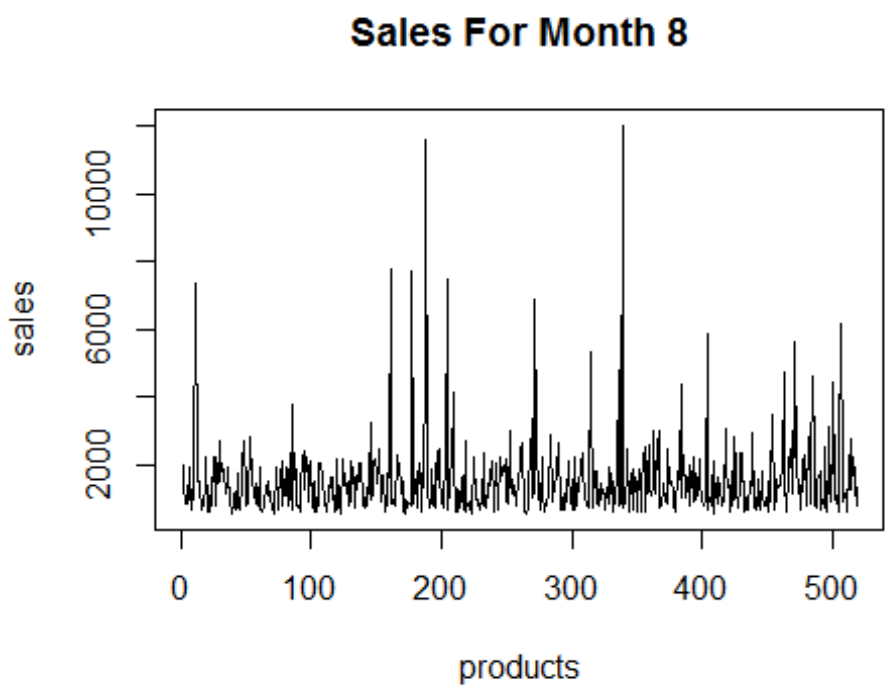

Sales For Month 3
sales
0
20000
40000
0
100
200
300
400
500
products

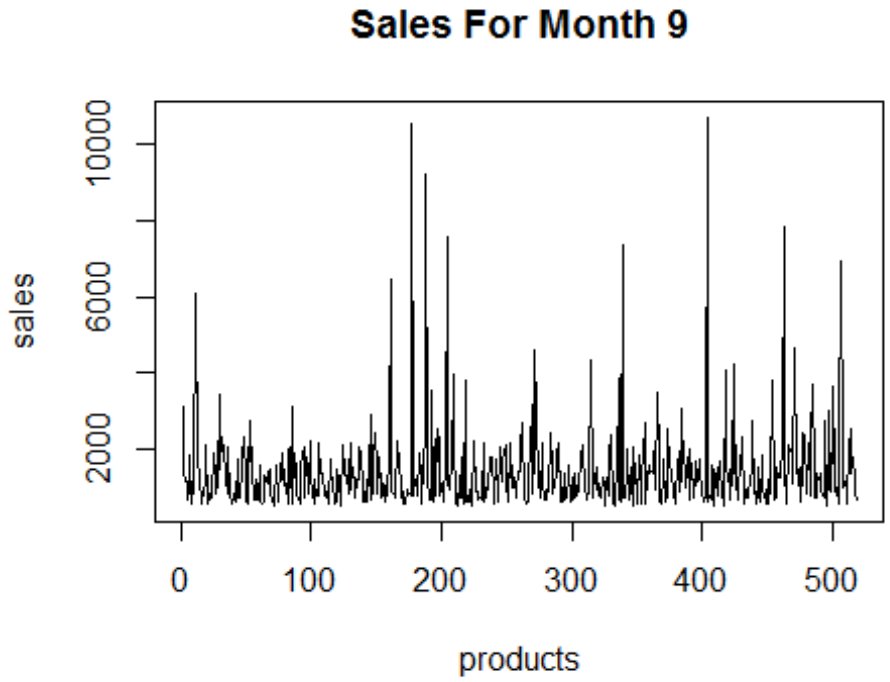

Sales For Month 4
sales
0
10000
30000
0
100
200
300
400
500
products

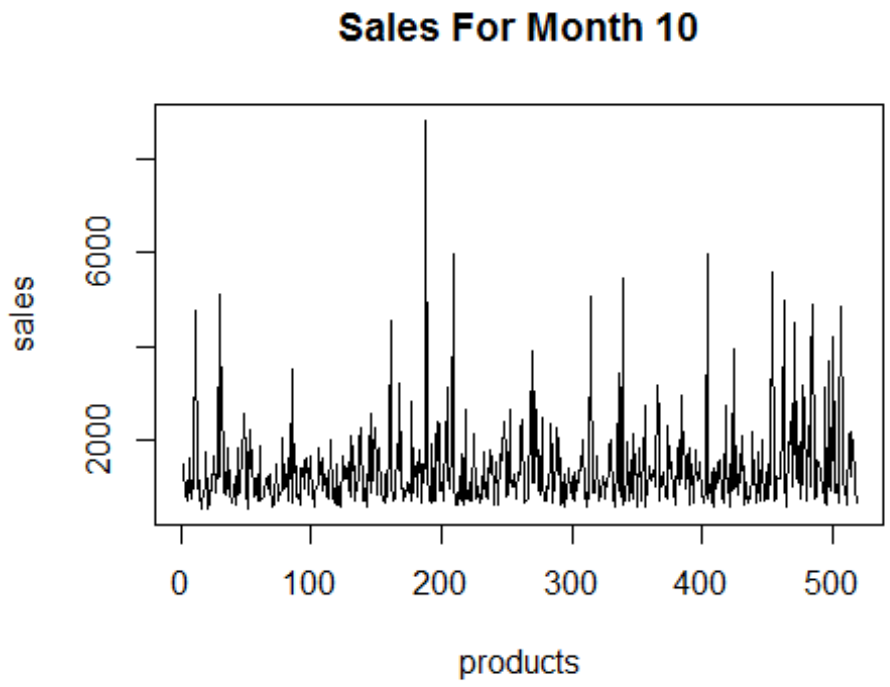

Sales For Month 5
sales
0
20000
40000
0
100
200
300
400
500
products

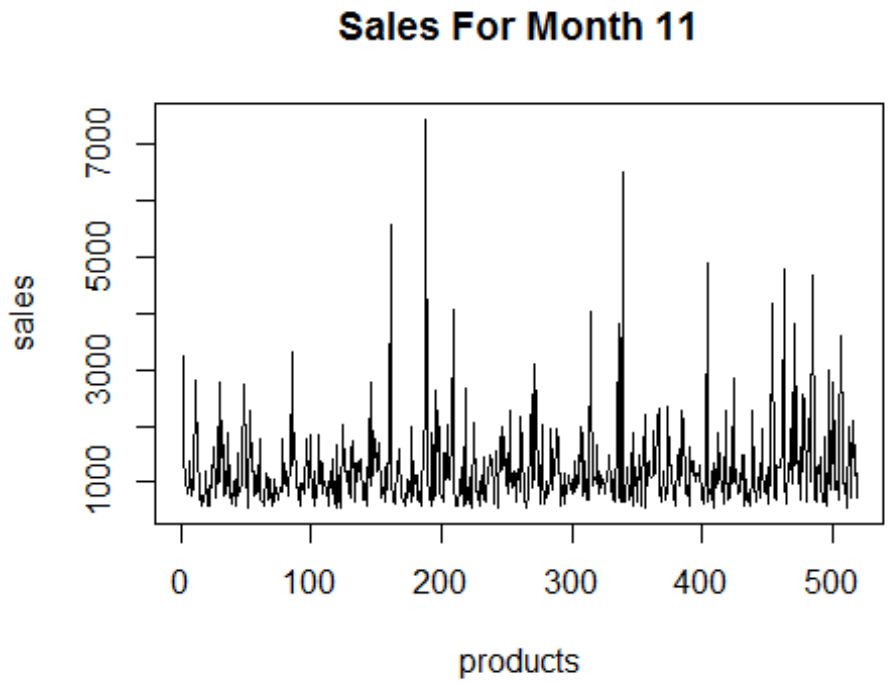

Sales For Month 6
sales
0
5000
10000
15000
0
100
200
300
400
500
products

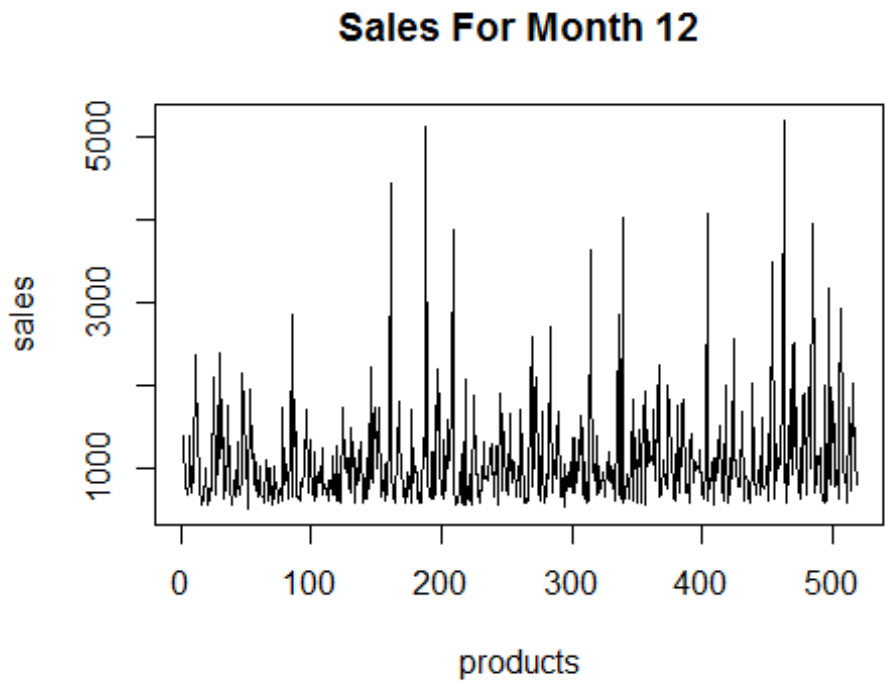

Sales For Month 7
sales
2000
6000
10000
0
100
200
300
400
500
products


Sales For Month 8
sales
2000
6000
10000
0
100
200
300
400
500
products


Sales For Month 9
sales
2000
6000
10000
0
100
200
300
400
500
products


Sales For Month 10
sales
2000
6000
0
100
200
300
400
500
products


Sales For Month 11
sales
1000
3000
5000
7000
0
100
200
300
400
500
products


Sales For Month 12
sales
1000
3000
5000
0
100
200
300
400
500
products